

February, 1982
NEWSLETTER



\$ 2⁰⁰

MICHIGAN ATARI COMPUTER ENTHUSIASTS

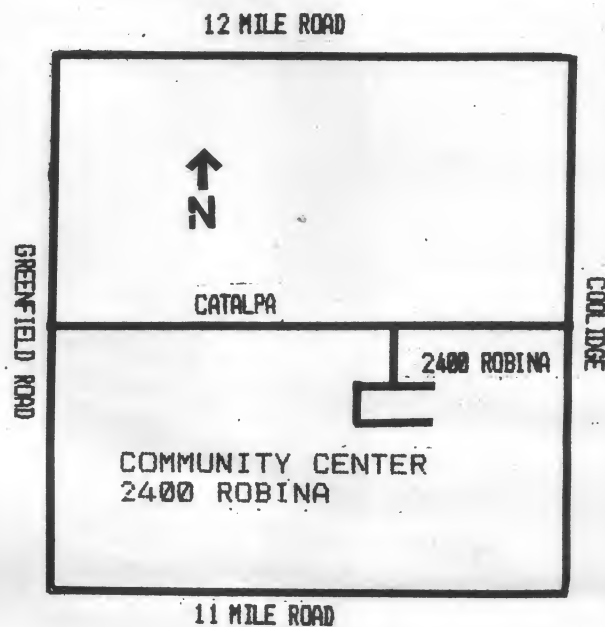
NEXT MEETING:

FEBRUARY 18

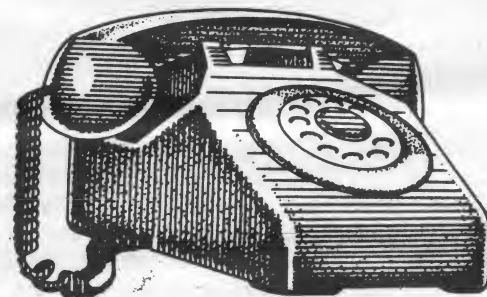
COMMUNITY CENTER
2400 ROBINA
BERKLEY, MICHIGAN

MEETING AGENDA:

6:45 OPEN SOCIALIZING
7:00 LIBRARIES OPEN
7:30 CALL TO ORDER
SHOW AND TELL
8:00 BUSINESS MEETING



IN THIS ISSUE...



BULLETIN BOARD
ONLY A PHONE
CALL AWAY!

SOFTWARE REVIEWS

CASSETTE I/O PART 2

DOWN MEMORY LANE!

BAKER STREET!



READY FOR A REAL CHALLENGE?



It's a battle of wits and fast action, this challenging game in ROM form for *Atari® 400™ or Atari® 800™ Personal Computer Systems. You're trapped in an Alien Fortress. Your mission is to clear your sector of the Alien Droids and progress to the next sector. The first sector seems easy enough...BUT-WATCH-OUT...as you progress from sector to sector, the Droids become more numerous and aggressive. A great challenge and fun too!

K-RAZY SHOOT-OUT is destined to be a favorite of Atari enthusiasts across the country with features like:

- Solid-state ROM pack
- Seven progressive levels of play
- Every game different
- Bonuses and penalties
- Full-color T.V. graphics and space-age sound
- Full-color instruction booklet
- Millions of variations
- A constant challenge

[WARNING: Playing the game could be HABIT forming]

Now at your Atari Dealer!

Coming soon:

- K-DOS™
- K-STAR PATROL™
- BUG-A-BOO™

*Atari is the Registered Trademark of Atari, Inc.

FOLD & TEAR

If not available in your area, send a check or money order directly to:
K-BYTE • P.O. Box 466 • 1705 Austin Avenue • Troy, Michigan 48066
 Toll Free Phone Orders: 1-800-521-0560 • In Michigan: (313) 524-9878
☐ \$49.95, plus \$1.50 shipping and handling. Michigan residents add 4% sales tax.
☐ \$1.00 for instruction book only.

NAME _____
 ADDRESS _____
 CITY _____ STATE _____ ZIP _____

Please allow 3 weeks for delivery.

CHARGE CARD NO. ☐ **VISA** ☐ **MasterCard**
 EXPIRATION DATE _____



SOFTWARE PREVIEW: MICROSOFT BASIC

by Craig Chamberlain 2/4/82

Ready or not, Microsoft BASIC for your ATARI is just around the corner. I recently had a chance to take a quick look at it and would like to share my observations with you. The purpose of this article is to show the new commands in Microsoft BASIC and point out the ways in which it differs from ATARI BASIC. I assume that the reader is familiar with ATARI BASIC.

Like ATARI PILOT, this offering will not be for everybody. This version of BASIC is very powerful and gives you a broad selection of new commands and statements. On the other hand, many of the nice features of ATARI BASIC are lacking. As always, there are tradeoffs. If you load Microsoft BASIC on a 32K machine you will have only 4K of free space left. Obviously 48K is necessary which is why ATARI is advertising this product for the ATARI 800 only. In the future ATARI hopes to release it in a 16K cartridge, so 400 owners won't be left out in the cold. But for right now, it is on a protected disk for \$89.95.

My personal opinion is that although Microsoft BASIC offers a lot more power, the beginner would be much better off by starting out with ATARI BASIC.

EDITING

No abbreviations are allowed, except the normal "?" for PRINT. In many cases it is necessary to enter space characters to separate statements from numbers. Examples:

```
L.10 illegal (gives a syntax error)
LIST10 illegal
LIST 10 legal
```

Also there is no error detection on line entry, so of course there can't be a special error token SAVED with your program in front of bad lines. There are some nice editing features, though. New commands include AUTO for automatic line numbering, DEL for line deletes, and RENUM to renumber program lines. You can specify a starting line number and offsets. The AUTO command tells you if the line you are currently entering already exists. The RENUM will renumber GOTO, GOSUB, etc. statements.

DEBUGGING

Unlike ATARI BASIC, changing any program lines automatically clears all the variables. Error code numbers are out, written error messages are in. CONT restarts a program right after the point where a STOP occurred, which is my preference, while ATARI BASIC also goes to the next line. TRAP has been replaced with ONERR and RESUME. The RESUME allows you to specify whether to continue where it left off, continue at next line, or jump to a new line. ERROR forces an error for debugging purposes. ERR returns the error number, ERL gives the line. TRON and TROFF provide trace functions during program execution by printing line numbers in brackets.

CHARACTER STRINGS

The normal Microsoft strings have been implemented. That means no DIM is necessary, and true string arrays can be used. The normal LEFT\$, RIGHT\$ and MID\$ functions are joined by INSTR and STRING\$. VAL gives a zero if the first character in a string is not numeric, whereas ATARI BASIC gives an error. There are tradeoffs. Not all characters are

continued

legal in strings (string literals within the program can't have a zero character in them). The maximum string length is 255 characters. The notorious garbage collection process is now required.

NUMBERS

Microsoft supports three degrees of precision: signed integer (two bytes); single precision (four bytes, 6 digits); and dual precision (eight bytes, 16 digits). The floating point numbers have a range from 1.70141E+38 to 5.87747E-39. I believe that literals are evaluated at run time. See a later section on arithmetic for speed comparisons.

VARIABLES

At least the first forty characters are significant. My understanding is that you are not limited to 128 variables. BASIC statements are always illegal as variable names. I don't think that BASIC keeps a variable list along with the SAVEed form of the program. Variables can be preserved and passed from one program to another with the COMMON command.

ARRAYS

You don't have to dimension for an array of ten elements or less. A DIM is illegal in the immediate mode. You can specify whether array elements start at zero or one. The commands DEFINT, DEFSNG, DEFDBL, and DEFSTR let you set default variable types.

LOGIC

The logical operators AND, OR, XOR, and NOT do true bit functions. Logical true is now -1, not 1.

COMMANDS

You can RUN at any line number. LIST uses a minus sign when giving a line number range. This range can be open ended. Examples:

```
LIST
LIST 10-90
LIST -90
LIST 10-
```

CLR is changed to CLEAR. You can also CLEAR the time STACK. Additional commands were explained in the debugging section.

FUNCTIONS

You can define functions but not in the immediate mode. The user-defined function can pass several arguments. The trigonometric functions allow only radians, no degrees. There is a TAN function. The USR function allows you to pass only the address and one integer argument. Since the argument is not pushed onto the stack, you have to retrieve it from page zero.

BREAK KEY

Hitting the BREAK key at any time prints the word "BREAK", sometimes with a line number. It also turns off all sounds.

FILE COMMANDS

The source storage and retrieval commands are LOAD, SAVE, CLOAD, CSAVE, LIST, MERGE and VERIFY. You can see that there are provisions for cassette owners. MERGE is actually ENTER. VERIFY will be welcomed by cassette users.

INPUT/OUTPUT

Microsoft seems to make it harder to use CIO. Instead of passing auxiliary arguments in the OPEN command you now have the words INPUT,

continued

BUSINESS AND HOME COMPUTERS SPECTRUM

THE COMPUTER STORE

WE OFFER:

- * A sales staff that gives personal service—and will answer all your questions.
- * Up-to-date selection of business and personal software.
- * A service department that cares for your equipment—in store or on site.



- * Complete line of computers, accessories, supplies featuring:
 - Altos Hard Disk/multi-user
 - Apple Computer Systems
 - Atari Computers

HOURS:

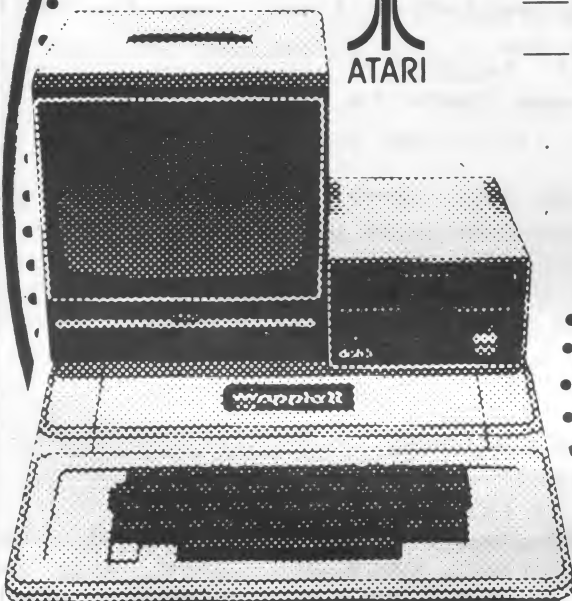
M-F 10 - 7:30 p.m.

Sat. 10 - 6:00 p.m.

SPECTRUM COMPUTERS

26618 Southfield Rd.
Lathrup Village, MI 48076
(Between 10 & 11 Mile)

Phone: 552-9092



OUTPUT, UPDATE and APPEND, for read only, write only, read and write, and write without erase. There is no XIO command. They do at least supply a special file called CIOUSR which contains common user routines. The new arrangement will be just fine for someone doing simple I/O, but there is still no direct tie to CIO for the ambitious programmer.

DISK COMMANDS

LOCK, UNLOCK, KILL and NAME were necessary because there is no XIO. KILL does a file delete and NAME is the rename command. NOTE and POINT are supported, but the POINT is actually a PRINT AT statement (see below).

PRINT

Semicolons behave differently than ATARI BASIC because Microsoft reserves room for a sign symbol. That means you get spaces between numbers separated by semicolons. A semicolon in the statement PRINT #6; is illegal and must be replaced with a comma. You do not, have to include a semicolon to separate different variable types. Example:

```
PRINT N"BYTES FREE" legal
```

Microsoft does have a TAB function and also a SPC function to print spaces. Cursor addressing is used in the PRINT AT command. Finally, PRINT USING is fully implemented.

INPUT

This command is illegal in the immediate mode. Leading spaces are ignored. You can specify a print line within INPUT statement, in which case the "?" symbol is not printed. It is possible to INPUT into an array. There is also a LINE INPUT statement which gets a whole line including commas and quotation marks.

READ, DATA, RESTORE

The READ statement allows reading into an array. Spaces in DATA statements are ignored. Double commas in a DATA statement make a zero, not an error. Quotation marks separate strings and are not actually read as part of the data. RESTORE allows a line number option.

FOR, NEXT

The NEXT statement does not require a variable after it if there is only one loop nesting. NEXT also allows a variable list, as in NEXT I,J,K. Example:

```
FOR K=0 TO 99 : NEXT legal
```

There is a limit to the nesting of loops, and there is no POP command.

IF, THEN

Microsoft also supports an ELSE statement on the same line as the IF and THEN.

GOSUB, RETURN

Subroutines cannot be nested indefinitely as in ATARI BASIC. Again there is no POP command.

GOTO, GOSUB

Variables are illegal in line number specifications.

GRAPHICS

You still have the commands GRAPHICS, COLOR, SETCOLOR and PLOT but you will have to get used to typing them out because there are no abbreviations. The DRAWTO does not exist because it is actually part of the PLOT. You can say PLOT X1,Y1 TO X2,Y2 TO X3,Y3 and so on. Some

continued

may prefer this over the old way. The new FILL command works in the same way. POSITION has been replaced with PRINT AT. LOCATE is done using SCRN\$.

SOUND

The only difference is an additional parameter at the end of the statement which is optional. You can now specify duration in terms of television frames.

CONTROLLERS

None of the joystick, paddle, lightpen or other controllers are supported but it won't be that hard to just PEEK the appropriate memory locations directly.

PLAYER/MISSILE GRAPHICS and CHARACTER SETS

Unfortunately there is no exciting news here. The only support is in the form of the RESERVE option which lets you set aside safe memory just for player/missile and character set buffers.

There are a few other commands not found in ATARI BASIC. RANDOMIZE allows sequences of random numbers to repeat if you so desire. The WAIT command monitors memory locations for changes; one of its few uses will be to detect when a vertical blank has just occurred. MOVE is a memory moving command that you must use carefully. The really neat new command is AFTER, which provides programmable software interrupts. Also there is TIME and TIME\$, which are used to set and display the time in hours, minutes and seconds. INKEY\$ is also supported.

ARITHMETIC

The arithmetic operators are the same, but the execution times are quite different. Microsoft uses its own math routines while ATARI BASIC uses the OS (operating system) math routines which are slow and have rounding errors. I ran several very simple benchmark tests on the two BASICs; the results are interesting.

```
FOR K=0 TO 9999 : NEXT K
```

The empty loop was done by Microsoft in 75% of the time ATARI BASIC took, which is 1.3 times as fast.

```
FOR K=0 TO 9999 : N=K : NEXT K
```

Now 78% but still 1.3 times faster.

```
FOR K=0 TO 9999 : N=K+K : NEXT K
```

This gave 80% but was still 1.3 times faster.

```
FOR K=0 TO 99 : N=2^K : NEXT K
```

A 17% here means that Microsoft is 5.9 times faster.

```
FOR K=0 TO 99 : N=SQR(K) : NEXT K
```

This gave 36% or 2.8. I then combined the four primary arithmetic operations as follows.

```
10 A=3 : B=44 : C=555 : D=6666 : E=77777
20 FOR K=0 TO 9999
30 N=((A+C)*(B+D)+(B-A)/(E-D))*K
40 NEXT K
```

The result is 64%, which means Microsoft completes this program in 64% of the time required by ATARI BASIC, so it runs 1.6 times as fast.

```
FOR K=0 TO 6.28 STEP 0.01 : N=SIN(K) : N=COS(K) : NEXT K
```

continued

Trigonometric functions were 27%, or 3.8.

My next project was to try the high level language benchmark in the September 1981 BYTE. This was a prime number program that used addition only. I liked this program because it contained GOTO, IF THEN and FOR NEXT statements. The problem was that I couldn't get the program to work under Microsoft. On a 48K system with Microsoft and DOS there are only 21020 bytes free! By changing an array dimension in the program from 8901 to 4501 I was able to test three BASICs: ATARI, Microsoft, and APPLESOFT.

LANGUAGE	TIME
APPLESOFT	152
integer Microsoft	156
single Microsoft	186
ATARI BASIC	203

Yes, the APPLE is faster but remember that no benchmark is totally fair. The prime number program didn't test multiplication, division, exponents, and so on. Also ATARI Microsoft has so many more commands and variable types than APPLESOFT that it has more to interpret. Only true applications will show which is better. As expected, Microsoft was faster than ATARI BASIC. Sheldon reports, however, that Microsoft seems to be much slower on some other things. Again it will depend on the application. The nice thing is that you now have a choice of two different types of BASICs. Those who want more power than ATARI BASIC but don't want to give up any of its features might want to take a look at BASIC A+ from OSS, the people who wrote ATARI BASIC.

One more comment. I was appalled at the inefficiency of the BASIC implementation of the benchmark. When I optimized it I used every trick I knew and was able to reduce it from 18 lines to 6. The optimized version under Microsoft BASIC using integer variables took only 120 seconds, which shows once again that a good system can be worthless if the programming is lousy. And, if you really want speed, you can still use a screen which requires less DMA or turn off DMA completely, something which APPLES can't do. Then the faster 1.8 Mhz speed of the ATARI will outstrip the 1.0 Mhz APPLE. Of course, the APPLE owners could buy a totally new microprocessor for their computer, but the key word is "buy".

In conclusion, those who have not had earlier experiences with a Microsoft BASIC are going to have their hands full. As you have seen, there are many differences between Microsoft and ATARI BASIC, with each one having strong points and weaknesses. Sheldon reports that this version of BASIC is very similar to the one used on Radio Shack models, so the biggest gain will undoubtedly be compatibility with other computers, which will open up a whole new world for ATARI. It's just further proof that ATARI is the way to go.

UNCLASSIFIED ADS

WANTED: Used 810 recorder
Invitation to Programming
Ken Van Dellen, 821-5706

MICHIGAN ATARI COMPUTER ENTHUSIASTS
IS IN NO WAY CONNECTED WITH THE
ATARI CORPORATION.

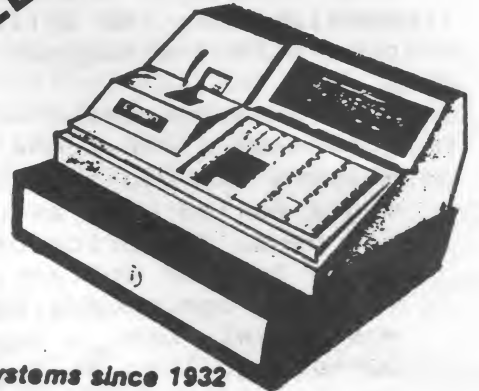
ATTENTION COMPARISON SHOPPERS

GRAND OPENING

CASH REGISTERS - SCALES - COMPUTERS
DISCOUNT PRICES!

RITE WAY ENTERPRISES 8262 12 MILE
WARREN, MI. 48093

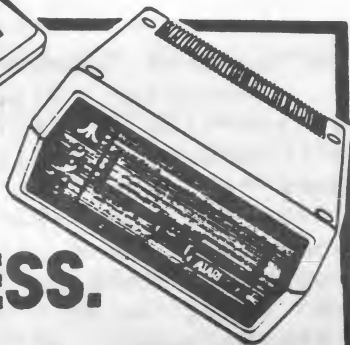
751-2454



rc allen

cash register systems since 1932

MODEL #	# OF DEPARTMENTS	LIST	OUR PRICE
2001	9 TOTAL	\$395	\$249
111A	6 DEPT.	\$795	\$495
2004	4 DEPT.	\$895	\$495
3004	4 DEPT.	\$995	\$595
3008	8 DEPT.	\$1095	\$695
3108/12	2 TO 12 DEPT.	\$1195	\$795
3016/48	16 TO 48 DEPT.	\$1395	\$895
9014/42	14 TO 42 DEPT.	\$1295	\$795
SUMIT SCALE		\$790	\$690



PAY LESS.

**GET
MORE.**

ATARI 800 with 16K memory \$749
ATARI 400 with 16K memory \$349
ATARI 810 Disk Drive \$439
ATARI 850 INTERFACE \$169

software 15% TO 20% OFF

EPSON MX- 80 \$479
100 \$759

HIGH PERFORMANCE \$1.99
FLOPPY DISK



**THE
DISCOUNT
Computer
SOURCE**

POKES FROM THE PREZ

Hello again! What a great turnout at the January meeting! We counted over two hundred and fifteen Atarians in our roomy new meeting place. I just hope we don't outgrow this one too! A big thank you to Star Commander Class One Ed Middlebrook for tipping us off as to its availability. We will have more monitors set up all over the room to cope with the copious crowds we've been attracting.

Some good news and some bad news about the program libraries: The good news is that the libraries will soon be available by mail to members who can't make the general meetings. Members who desire their media mailed will assume the cost of postage and packaging. Now for the bad news. The price of library cassettes may have to be raised from three to four dollars. Our tape librarian could conceivably pass on from old age producing tapes for almost three hundred people! We will most likely use a duplicating service instead of running them off ourselves. This will also give us more time to work on documentation and instructions for the programs. MACE still charges far below the norm for its library media...a survey of other groups indicated an average price of eight dollars.

I recently looked at some old reviews of software for our machines and I couldn't help reflecting on how much more discriminating the quality of the new software being released these days has made us. I remember in the olden days (1980-81!?!) when hungry wolf packs of Atarians would lurk in the shadows of local computer stores waiting in ambush for unsuspecting UPS drivers. These rogueish creatures would eagerly devour soft goods the like of which would seem sorely unpalatable to those of us in these civilized and genteel times. A great program for 1981 may be mediocre in comparison to what the year has yet in store for us. Keep that in mind when you read older reviews.

At the risk of being hamstrung by the general membership, we are trying out something new in the way of newsletter distribution. I'm sure everyone liked the new expanded magazine-like format, but nobody likes not knowing when its coming. The February newsletter will be distributed to all paid-up MACE members at the February general meeting. The members who don't attend will be mailed their issues the following day. If this is odious to your ears and incites you to outrage, let us know. We can be verrry flexible. Remember, you already know the meeting schedule for the entire year; ergo, you know when you'll get your newsletter. There's very little danger of us being late this way...do you honestly think your officers would show their faces to a crazed mob on a third Thursday without the coveted MACE Newsletter?

The MACE A.M.I.S. (Atari Message and Information System) Bulletin board is currently up for MACE modem users from 9 PM thru 12 Noon daily on (313)-868-2064. We will be up as the first 24 hour-a-day Atari based bulletin board, as soon as all of the dedicated equipment necessary is together. Atari User Group Support was so impressed with the prototype system that it is making AMIS the standard Atari Bulletin Board System for User Groups. Kudos to AMIS Lead Programmer Tom Giese and all of you who helped us by calling in to the several test systems we put up over the past months. Watch for a feature article from the Communications SIG about AMIS in the coming months.

Both the program libraries and the newsletter need contributions from YOU. The newsletter in particular can really use articles from you beginning, intermediate and advanced programmers. Share what you learn about your machine with the other people who are asking the same questions that you have found answers to. Software and hardware reviews from members are also welcome. You don't have to write well to be published...I'm certainly proof positive of that!

The MACE Birthday Party will be held in April or May at a regular membership meeting. Atari Incorporated, having recognized MACE as the largest User Group in the world will be visiting us, and we plan numerous festivities for a supportive club membership that has made MACE the acknowledged leader in its field. ADMITTANCE TO THE BIRTHDAY PARTY WILL BE LIMITED TO PAID-UP MEMBERS ONLY, SO DON'T FORGET TO BRING YOUR MACE CARD!

See you soon,



Arlan
Micronet: 70675,463
Source ID: TCT987

BULLETIN

SPRING BASIC CLASSES

Basic classes for beginners will be offered starting in March. The MACE classes provide an excellent exposure to the basics of programming your ATARI personal computer. The courses also include an excellent introduction to the graphics and sound capabilities of your computer system.

No prior programming experience is required, although it is advisable to read through the materials packaged with your ATARI, especially the programmed text on ATARI BASIC by Albrecht, Finkel, and Brown.

These classes are an excellent way to start getting the most out of your computer as quickly as possible. But remember, the classes fill up FAST! If you are interested, contact:

Mark Davids
774-9709

EDUCATION GROUP NOTES

The February 2 meeting was cancelled due to the snow conditions. Call Mark Davids for information about the next meeting. 774-9709

JANUARY MEETING MINUTES

Sheldon Leemon, Secretary

When I arrived for the January meeting and saw the cavernous hall, I thought that at last we had some room for expansion. That thought lasted about half an hour, until over 200 people had arrived, about half in attendance for the first time. Hmm...100 new members a month, times twelve months, carry the Apple owner,...Yipes! Cobo Hall for our '83 meetings?

For Show-and-Tell, yrs truly gave a modest account of the Atari Star Awards banquet, for which he was flown to semi-sunny San Francisco by the guys who wear the Fuji "A". Although Instedit did not win the top prize, I did get to sit next to Fernando Herrera at dinner (he got the 25 big ones for MY FIRST ALPHABET). The other finalists were Greg (too young to be rich) Christensen, author of Caverns of Mars, and Ronald and Lynn Marcuse, authors of Data Base Mangement, Weekly Planner, et al. Those who saw me demonstrate the finalist's programs at the meeting could see that they were of comparable quality to the best on the commercial market today. ALPHABET is a terrific hi-res alphabet book that teaches toddlers their A,B,C's with such style that parents'll have to wrestle the tykes to use the 800. Caverns of Mars is a spectacular implementation of an arcade favorite. This program is so hot that Atari has made it their own, and will be moving it to the main catalog, along with ALPHABET. For now, they are still available for less from APX. Having met the other finalists, I must compliment Atari on their decision. Clearly, Fernando was the best choice for the Superstar award. Besides being a super nice person, the many loving hours he devoted to writing the program really show.

Although only shown casually before the meeting, Gary Luzier's trackball caused quite a stir. Be sure to let us know when you can produce a few to sell, Gary. Missile Command will never be the same. In a surprising development, a request from the Prez for equipment for a club CBBS (bulletin board), met with the offer from a couple of members to supply same gratis. Further homage to our heroes will appear as soon as they deliver.

For the main program, we finally go around to forming committees. A complete listing of these special interest groups, and their leaders will be printed elsewhere in the newsletter. Here's your opportunity to meet with others on a more personal basis, and to help develop our resources. You can get more information on these groups by coming to the next general meeting. Remember, from now on, meetings are definitely the third Thursday of the month, at 7:30 P.M. at the beautiful Berkley Community Center. See you there.

DOWN MEMORY LANE

by Sheldon Leemon

Here we are again, seekers of truth and wisdom. Better get on you boots for this month's stroll down the lane, because we'll be leaving behind the safe lanes of Appendix I of the Reference Manual, and getting into the really deep stuff. Today's topic deals with the mechanics of the screen display.

The graphics feature which we'll be exploring is display memory indirection ("Funny, the words sound like English, but they don't make any sense."--Read on.). Other microcomputers use a fixed area of memory to hold the data being displayed on the screen, with each byte corresponding to one character position on screen (the famous "one byte, one vote" principle). This means, among other things, that in order to move images on the screen, you must move the screen data around within this area of memory. The Atari computers, however, do not use a fixed area for display memory. Instead, two bytes of memory act as a pointer to display memory. By changing this pointer, it is possible to change the data being displayed without moving around any bytes in memory! You may think of this system as keeping a "window" on memory. Instead of moving the contents within the window, the frame is moved over the contents. But we're getting ahead of ourselves here. This topic, known as coarse scrolling, will be grist for next month's mill.

Our current avenue of inquiry has to do with the fact that two sets of pointers are kept. While one tells the ANTIC chip where the data being displayed is located (I call this the "display pointer"), the other tells the Operating System where to store data which is entered from the keyboard or PRINT statements (the "write pointer"). While these will normally have the same value, so that the display shows what is written, it is quite possible to print words or graphics into one part of memory, while displaying another part.

The write pointer is contained in memory locations 88 and 89, known in OS parlance as SAVMSC. You can verify this by calculating the address pointed to with the statement `WP=PEEK(88) +256 *PEEK(89)`, and then using a `POKE WP,33` command to make the letter "A" appear in the upper-left corner of the screen (screen data uses Internal Character Set values, shown on page 55 of the Reference Manual). At this point, the write pointer is the same as the display pointer, so what you write is what you see. Now for the \$64,000 question. Why would you ever want to write data to memory locations where it won't be immediately displayed?

continued

One reason to change the write pointer is to allow to you set up another display area while the user is watching the current display. Then you can switch to the other display instantly, without the user having to watch it being set up. This technique is sometimes referred to as page flipping. A sample program demonstrating this technique is set forth below. While the REMark statements pretty well describe what is going on, I will explain the first line in a little more detail. The GR. 24 (=GR.8+16) statement is used merely to clear the top 8K of memory, so that random characters won't be displayed when we flip pages. Next we take the value of the top-of-memory pointer (PEEK 106) and lower it by 4, so that the computer now thinks that the top of memory is 4 pages (1024 bytes) lower than it actually is. Then, when the GR. 0 command is given, the computer puts the display list and display memory 4 pages lower in memory than normal. This leaves the top 4 pages open for us to set up our new display area. All we have to do then is take the write pointer, alter it, write in the new area, and flip back and forth by changing the value of the display pointer to its original value, or its original value + 4. Like so:

```
0 REM ** PAGEFLIP DEMO by Sheldon Leemon
2 REM **
8 REM ** Set aside 4 pages of memory for second screen,
surpress the cursor
9 REM *
10 GRAPHICS 24:TOP=PEEK(106):POKE 106,TOP-4:GRAPHICS
0:POKE 752,1
15 REM *
16 REM * DMEM=Pointer to display memory, WMEM=Pointer to
memory written
17 REM *
20 DMEM=PEEK(560)+PEEK(561)*256+5:WMEM=PEEK(89)
25 REM *
36 REM * Next three lines set up the display on both
screens,alternating the page of memory being written
37 REM *
40 DIM A$(756),B$(756): A$=CHR$(18): A$(756)=A$: A$(2)=A$:
B$=CHR$(124): B$(756)=B$: B$(2)=B$
45 POKE 83,37: POSITION 2,1: ? A$: POKE 89,WMEM+4:
POSITION 2,1: ? B$
50 FOR I=1 TO 2:POSITION 4,23: ? "HIT ANY KEY FOR A QUICK
CHANGE ";CHR$(28): POKE 89,WMEM: NEXT I
55 REM *
56 REM * Flip routine
57 REM *
60 POKE 764,255: REM ** Reset keyboard register
64 REM *
65 REM ** If key is struck,display other page
66 REM *
70 IF PEEK(764)<255 THEN POKE DMEM,WMEM +4*(PEEK
(DMEM)=WMEM): GOTO 60
80 GOTO 70
```

continued

You might also like to take note of the method used in line 40 to fill the strings with 756 repetitions of the same character. Try it yourself, it works! Now, if you change line 70 to read POKE DMEM, WMEM +4*(PEEK (DMEM)=WMEM) : GOTO 70 the display will continuously flip, giving you a kind of flickering checkerboard effect. This may give you an idea of how page flipping can be used to create a limited form of animation.

I will close by saying that this excursion into the mystic realm was prompted by a reader's inquiry. Is there some aspect of the Atari's inner workings that is keeping you up at night? Perhaps the veil can be pierced for you, too. Write, care of this station.

```

0100 ;GETTING RID OF AN ANNOYING PROBLEM
0110 ;           By Edward Chu
0120 ;For those of you who are thoroughly annoyed
0130 ;with the ATARI ASSEMBLER EDITOR's loud
0140 ;response to an error, here is a program to
0150 ;disable the BELL character's noise. To use
0160 ;this program, enter DEBUG and type:
0170 ;C 346<EF,6
0180 ;The address in $346 is one less than that of
0190 ;the routine. For convenience you may want to
0200 ;make this an AUTORUN.SYS and/or change the
0210 ;origin. This program is only 12 bytes long
0220 ;and can go just about anywhere.
0230 ;
0240 ;Note: this program is not SYSTEM RESET proof
0250 ;Note: If anyone cannot get this to work call
0260 ;      me at 646-0792 as a last resort
0270 ;
00FD 0280 BELL    =    253
02A2 0290 ESCFLG =    $2A2
F6A4 0300 EOUTCH =    $F6A4
0000 0310          *=$6F0
06F0 C9FD 0320          CMP #BELL
06F2 D005 0330          BNE OK
06F4 A280 0340          LDX #128
06F6 8EA202 0350          STX ESCFLG
06F9 4CA4F6 0360 OK      JMP EOUTCH
06FC          0370          .END

```

HIGH SCORES

MISSILE COMMAND

RICHARD GIZYNSKI 730,000

JAMES HETZEL 499,855

EASTERN FRONT

RICHARD GIZINSKI 255!

LET US KNOW YOUR SCORES!

H C O M P U T E R H O R I Z O N S

JANUARY IS COMMUNICATIONS MONTH



ATARI 830™ Acoustic Modem

REG 564⁹⁵

SALE 450⁰⁰



ATARI 850™ Interface Module



THE SOURCE
AMERICA'S INFORMATION UTILITY



(313) 464-6502

37099 SIX MILE ROAD • LIVONIA, MICHIGAN 48152

NEW PRODUCT RELEASE

MORE ATARI MEMORY

Mosaic Electronics has just announced the 16/32K RAM board, which is expected to be their most popular product to date. The 16/32K RAM adds 16K to an ATARI computer system. After the ATARI user has exhausted the potential of 16K, upgrade to 32K is very easy, using a separate \$60 upgrade kit. Atari 400 owners can use their existing 16K RAM to upgrade to 32K for \$120 total. The Mosaic 16/32K RAM is of particular interest to owners of the Atari 400 with 16K or 800 users with 16 or 32K configurations. For more information write to Mosaic Electronics, P.O. Box 748, Oregon City, Oregon 97045.

CORRESPONDENCE

25 Jan 82

Craig Chamberlain
% M.A.C.E. Newsletter

Dear Craig:

This is an open letter to you. By that, I mean that I hope you will allow it to be published in the newsletter, but I leave that choice in the hands of you and the editor. I probably wouldn't be writing to M.A.C.E. again so soon except that I had to disagree with you again. Where I disagreed slightly on a few points in my last letter, I must absolutely tell you that you are DEAD WRONG this time.

With such an introduction, I hope your competitive juices are stewed up. Allow me to tell you where you are wrong: you called yourself a jerk. Nuts! I just wish that half the programmers in the world (which equates to maybe one-half of one percent of all people?) were as "jerky" as you. The only thing you had right about yourself was saying that it took articles like yours to make people like me write letters. TOO TRUE! If your article hadn't been well written, thought provoking, and loaded with examples, I wouldn't have read it. And if I hadn't read it, I wouldn't have even thought about some of the ideas you presented!

There is a common conception that program authors are the best source of information about their product(s). That is true ONLY if you know what questions to ask! Many times people have asked me how to write faster and/or more efficient programs, but NOT ONCE, until I read your article, did I consider the uses of ON...GOTO !

Anyway, I hereby declare for once and for all that Craig Chamberlain is NOT a jerk. And articles like "Hyperwarp..." and "RTANGLE" are good enough that the only thing that M.A.C.E. has to worry about is whether Craig will continue to write for them or go "pro" for the big bucks (well...would you believe little bucks).

One last comment: Craig and I still don't see eye-to-eye about GOingTO a REMark statement, but our viewpoints are different: I seldom write games, etc., where space and/or time is critical; instead, I write big, cumbersome programs where maintainability is of primary importance. Even so, I went back and looked at a few of my game-style programs; sure enough, in many places I had placed REMarks just ahead of the GOTO target lines, just as Craig suggested. Score two for Craig. (But if the truth were known, I probably wrote the program without REMarks and went back and added them as best I could. Naughty, naughty.)

And now, just so Craig will know that I wasn't picking on him, I would like to comment on Richard Gizynski's "Baker Street Bytes" in your January issue. There is an easy way to fill a string with a single character (or sequence of characters). Actually, this method has been around since we wrote BASIC, but for some reason Atari has never (until "De Re") really publicized it. Anyway, I would like to rewrite lines 85 through 150 of Richard's listing:

```
90 MAINSTRING$=" ":MAINSTRING$(SIZE)="X":  
    MAINSTRING$(2,SIZE)=MAINSTRING$  
100 NAME$=MAINSTRING$  
110 NUMBER$=MAINSTRING$  
120 LINE$=MAINSTRING$  
130 CLEAN$=MAINSTRING$
```

The trick is that string LETs proceed from start to finish a character at a time, rippling if necessary. Want to fill NUMBER\$ with "12345"? Try this: NUMBER\$="12345":NUMBER\$(38)="X":NUMBER\$(6,38)=NUMBER\$

A warning: remember that there is a bug in Atari BASIC (but not in BASIC A+). Assigning (LETting) a string such that BASIC needs to move an integral multiple of 256 bytes will not work. To assure that your program doesn't fall into this trap, always move an odd number of bytes (if you need an even number, move it in two chunks). (Of course, you needn't worry about this if you are using strings DIMensioned less than 256.) An example:

Suppose we want to add a "Delete old entry" command to Richard's program. Add the following lines:

```
1095 ? "9. DELETE AN ENTRY"
1135 ON (A=9)+1 GOTO 1000,2200
2200 REM DELETE AN ENTRY
2210 ?"}WHICH ITEM TO DELETE ? " ;:INPUT ITEM
2220 IF ITEM<1 OR ITEM>=(SIZE/38) OR ITEM<>INT(ITEM) THEN 1000
2230 I=ITEM*38+1
2240 MAINSTRING$(I,I+37)=MAINSTRING$(I+38,SIZE-1)
2250 MAINSTRING$(SIZE-38)=MAINSTRING$(SIZE)
2260 MAINSTRING$(SIZE-37)=CLEAN$
2270 GOTO 1000
```

Note line 2240: the use of SIZE-1 ensures an odd number of bytes will be moved. Line 2250 moves that last byte in safety, and line 2260 fills the gap at the top of the string with blanks. The I+37 in line 2240 is necessary; otherwise, BASIC will think the string should have its length adjusted. (I have NOT tried all this in this particular program, but it should work.)

I really should quit now, but I noticed one other trick I can tell you about. Look at lines 760-790 (for example). Notice all the usages of NAME\$(1,1). Why not do the following:

```
77 DIM ANSWER$(1)
760 INPUT ANSWER$
770 IF ANSWER$="M" THEN 970
780 IF ANSWER$="T" ... etc.
```

The point being, of course, that INPUTting into a one character string is guaranteed to produce at most a one character result! And if I can get in a small plug here, I would like to show the BASIC A+ equivalent of lines 755 through 790 (lines 780 and 790 get deleted):

```
755 input "Tape/Disc or return to Menu ? ",answer$
760 on find("MTD",answer$,0) goto 970,840,800
770 goto 740
```

We could handle lower and upper case answers via the following:

```
760 on find("MmTtDd",answer$,0)/2 goto 970,840,800
```

By the way, I really should congratulate Richard on his article. It manages to compress a usable program into an article-length space, something which I KNOW is hard to do. (And his program is remarkably similar in concept to several in-memory data filing systems now being sold, so keep it for reference.)

My schedule is so overloaded now that it is hard for me to find time even to write my articles for Compute! The fact that I find myself writing to your newsletter twice in three months says something about its interest value and the quality of your writers. Wouldst that some of the Compute! articles could measure up to your standards.

Bill Wilkinson,
Optimized Systems Software

BINARY

BINARY

BINARY

BINARY

BINARY

SUPPORTING



WITH

SOFTWARE

THIS MONTH ONLY !!

- CYPHER BOWL -- The FOOTBALL GAME that puts you right in the middle of all the action! Multiple offensive and defensive plays. Pass action, blocking, punting, and much more. Requires only 16K and comes on cassette and disk.
- | | |
|------|---------|
| Reg. | \$49.95 |
| NOW | \$39.95 |
- JAWBREAKERS -- THE ORIGINAL (the one Atari sued ON-LINE about) On Disk only. Bound to be a collectors item. Copies were selling across the country for over \$70! Limited Quantities. Buy yours now!
- | | |
|------|---------|
| Reg. | \$29.95 |
| NOW | \$24.95 |
- SOFT-PORN -- A REAL FUN ADVENTURE GAME by ON-LINE that uses a little graphics and a lot of wit. For adults only. Extremely mild. Great fun at a party! If you like adventure games you must try this one. On disk only. Needs 40KB.
- | | |
|------|---------|
| Reg. | \$29.95 |
| NOW | \$24.95 |
- ADVENTURES -- SCOTT ADAMS has 12 ADVENTURE GAMES just for you. They get progressively harder and more interesting as you get towards the higher numbers. A lot of fun. Very challenging. You need 24KB. Our sale for cassettes only.
- | | |
|------|-----------|
| Reg. | \$19.95ea |
| NOW | \$14.95ea |
- DATASOFT -- ALL DATASOFT SOFTWARE (Yes, including the very popular Text Wizard) is on sale for 10% off. Try your hand at programming in LISP, play with the DISK DETECTIVE, and the CHARACTER GENERATOR. We have all their stuff here for you to look at.

also, if you own a disk drive, don't miss this one!!!

BASF DISKETTES - A Box of 10 Regular \$34.95 NOW \$23.95

HUNDREDS OF PROGRAMS IN STOCK !

BAKER STREET BYTES
FEBRUARY 1982
By Richard Gizynski

Last month, in the program Telephone Directory, I showed how to concatenate (join together), make a part of a string appear flush right, bubble sort (one way of alphabetizing), and save and retrieve the string. One small bug appeared in the printing. Line 90 should have an end quotes (") after forty spaces. Unfortunately, printers set to approximately 40 characters wide, with a lot consecutive spaces in the line, have trouble printing it out.

Actually you only needed 38 spaces since the string was dimensioned to 38 spaces. This brings us to the subject of this article for this month. String lengths and sorts and how Atari handles them.

First the strings. Atari reserves space at the end of the program to hold its DIMensioned strings. When you have an INPUT, or write a STRING\$, Atari opens up the DIMed string. If the input is larger than the string, only the part equal to the DIMed portion of the string will be taken.

```
10 REM STRING AND BUBBLE SORT DEMO
20 REM BY RICHARD GIZYNSKI
30 DIM STRING$(38), CLEAN$(38),
```

```
    MAINSTRING$(1800)
40 REM OPEN AND TEST CLEAN$
50 REM TRY TO INPUT 38 SPACES AND 2 X'S INTO CLEAN$
60 CLEAN$="(38 spaces)XX"
70 PRINT CLEAN$
```

If you run the program at this point, Atari will print a blank line. The string CLEAN\$ has only a DIMension of 38.

Next we set up a simple INPUT and join the inputs together to form a large string.

```
80 REM INPUT AND CONCATENATE
90 PRINT "INPUT A NAME"
100 PRINT "IF YOU WISH T= STOP--INPUT
    STOP"
110 STRING$=CLEAN$
120 INPUT STRING$
130 IF STRING$="STOP" THEN 170
140 STRING$(38,38)=" "
150 MAINSTRING$(LEN(MAINSTRING$)+1)=
    STRING$
160 GOTO 90
```

On line 110, STRING\$ is made equal to whatever portion of CLEAN\$ it can hold. It also prevents leftover garbage in memory from being put into your MAINSTRING\$. On line 120, STRING\$'s length is changed to the length of the INPUT. Line 140 opens up STRING\$ to its full DIMed length

continued

by using the double subscript (38,38) to tell Atari that the character that starts at position 38 and the character that ends at position 38 in STRING\$ is a blank. If you leave out this line, the equal length segment of MAINSTRING\$ would not be there.

Line 150 is saying "find the current length of MAINSTRING\$ -- LEN(MAINSTRING\$) -- and, starting with that length plus one, add STRING\$ to it." It also says "make MAINSTRING\$ end where the added section ends. Including the blanks at the end of STRING\$." When the program is run, MAINSTRING\$ will hold up to 50 names.

NOW on to the sort.

```

170 REM BUBBLE SORT ROUTINE
180 X=1
190 IF X=0 THEN GOTO 290
200 X=0
210 FOR F=1 TO LEN(MAINSTRING$)-38
    STEP 38
220 IF MAINSTRING$(F,F+37))
    MAINSTRING$(F+37,F+38+37) THEN
    GOTO 240
230 GOTO 280
240 STRING$=MAINSTRING$(F)
250 MAINSTRING$(F,F+37)=
    MAINSTRING$(F+38,F+38+37)
260 MAINSTRING$(F+38,F+38+37)=STRING$
270 X=1
280 NEXT F
290 GOTO 190
300 PRINT MAINSTRING$
310 END

```

Line 180 sets the variable X to 1. Line 190 tests to find out if X is 0 to find out if the sort is complete. Line 200 sets X back to 0. If the FOR-NEXT routine that follows doesn't send you to the name switching segment, you're done. Line 210 sets up a FOR-NEXT routine equal to the number of names in MAINSTRING\$.

Line 220 checks to see if the segment of MAINSTRING\$ that starts at F and ends 37 characters down is larger than then segment beginning at character 38 and continuing to character 38+37. If you checked for smaller than, the program could run till you hit break if two adjacent names were the same. It sends you to the sort routine if name your at is 'larger' than the next name in line.

Line 230 is a GOTO by itself because if the FOR-NEXT loop is ended with two names that are in the correct position, the sort routine would switch them.

Line 240 takes advantage of the fact that since STRING\$ is DIM'ed to (38), it will pick up only 38 characters

continued

BAKER STREET BYTES

starting with the one at position F. Though STRING\$ was used earlier as an INPUT, now we use it as temporary storage.

Line 250 puts the name that was in position 2 into position one. Here and in line 260 MAINSTRING\$ MUST have the starting and ending positions defined or Atari will think that you just shortened it. Line 260 puts the original first name in the second spot.

Line 270 is used to tell line 190 that you have gone through the sort routine. It could have been put on line 260 but never on line 280 as line 280 sends you back through the FOR-NEXT loop. After all the FOR-NEXT loops, line 290 sends you back to line 190 to check to see if anything was moved. If it was, back to the FOR-NEXT loops.

Line 300 is why I set the string segments to 38 characters. It allows MAINSTRING\$ to print out on the screen with the names flush left since the screen is, by default, 38 characters wide.

One additional note. If you take out line 130 after having entered a longer program or using names of varying length, you will see a lot of strange characters. These are either the leftover tokens that Atari uses when it runs the program or characters that were used in longer names.

COMPUTER CENTER

The Computer Store for EveryOne

West Bloomfield
4381 Orchard Lake Road

SALE ENDS FEB. 28TH

Garden City
28251 Ford Road

855-4220

810 DISK DRIVE (1 ONLY)

425-2470

ONLY \$425.00

or **422-2570**

WITH THIS AD

3M 5 1/4 INCH DISKS (744-0) BOX OF 10 \$32.50

850 INTERFACE (1 ONLY) \$150.00

ATARITM

REVIEW: THE ATARI PERSONAL FINANCIAL MANAGEMENT SYSTEM

by Ed and Sharon Middlebrook

It was with great delight that we agreed to test the Atari Personal Financial Management System. This was a product that we had been waiting for a long time. As much as we enjoy Star Raiders and Missile Command, it was time to do something "useful" with our system.

The Personal Financial Management System (PFMS for short) comes with documentation about the size of the Basic Reference Manual in an attractive, indexed binder. Undoubtedly, a fair portion of the \$75.00 list price went into packaging this product. The PFMS comes with two program disks and a data disk.

The first chapter was designed for someone who had just purchased his/her Atari along with the PFMS, and needed guidance on plugging in the disk drive, turning on the power, etc. We feel this chapter should have been left out of the PFMS documentation. The rest of the manual is fair, but bounces between the obvious needs of a home user of the system and an accountant. It spelled out the technique of adding checks and deposits to the database, but then glossed over the important but vague topic of the Transfer file.

The basic premise of the system is great. You define budget categories to suit your needs. The categories may be unique, such as Groceries, or may be further subdivided, such as Home-Mortgage and Home-Insurance. Then, as checks are entered into the database, you specify which budget category applies. The amount of the check is automatically deducted from the budget specified. At any time, a detailed report on the budget categories may be generated, either as specific data or as a bar graph. Once a database of several months is built, forecasts can be projected for up to 30 months.

A very nice checking and savings account record keeper is included. Checks are entered by number, name, amount, date, category, and remarks. It is then possible to print a listing of the checks by almost any field, category, or number range. The listing can either go to the screen or a printer. Deposits and automatic withdrawals are handled in a similar manner.

The Transfer file is a handy feature, once you figure out what the documentation is trying to tell you. In general, it is for transferring money from one account to another so that a credit card bill for a budget category can be paid by a check without causing a double deduction from that category.

We found several minor irritations and one major fault with the PFMS. The system is slow in spots. It takes 30 seconds to 1 minute to enter each check. The update processing which takes place after all transactions are completed takes several minutes. The copy function, which backs up the data disk, is obviously a sector copy of the entire disk, and is painfully slow. It took six pairs of disk swaps and about 13 minutes to complete the backup procedure. This compares poorly to 3 swaps and 6 minutes using the Superdup disk copy from Micro-Media magazine or the MACE sector copy program by Ed Schultz.

continued

Atari designed the system to run on a single drive, with prompts for switching between the data and program disks. However, the documentation states that the PFMS will ONLY run on one drive. If you have a two drive system, you can't put the program disk in one drive and use the other for data. You MUST run with only one drive and live with all the disk swapping.

The major fault we found with the system is simple -- IT DOESNT WORK PROPERLY!!!!!! We conducted our tests with two different PFMS packages, and both blew up during processing. The system typically accepts all the inputs and processess them without any errors being indicated. However, when you go back to clear a returned check, edit a field, or just print a check, you eventually will get a message "UNEXPECTED ERROR - CONSULT OWNERS MANUAL" after processing a few months worth of data. This happened without fail after we input about two months of data. Only once did we get through two months, and then it bombed on the third month. The owners manual really shines in the subject of errors -- a paragraph mentions that if an error occurs, you had better have a backup copy, because you just lost everything on that data disk. Thats it -- no other help.

We finally managed to speak to a Technical Support person at Atari, who admitted they have been receiving calls from all over the country about failures in the system. At the time of our call, he had no idea what the final outcome would be. He speculated that either a new release would be issued or a sheet of fixes would be sent out to all owners of the PFMS, depending on the complexity of the fix.

At this point, we simply cannot recommend the PFMS under any circumstances. If and when Atari fixes the programs, it will be a worthwhile addition to almost anyone's library. As it stands, it rates a straight "F" for failure and a caveat to any buyer.

WORD SEARCH PUZZLE

V C E O B U S S E E O Q R E S E T C I F
S W R D A L A R E H P I R E P U Z K E G
B K U A O W K S E R Z I Q W M J W G X N
N K H T J M L B S B J B M Y H S C N A I
S V B A O T P C C V Q V A R M I S W C X
P N L D Y R O M E M Q S R F E S E R O E
A V R L O C I C A L F D G Y T Y C J N L
C Z K A J E D X I E P C O C S L T Q T P
E P A C H N E N L Z D W R R Y A O O R I
Y I E R Y I L O T I R O P G S N R C O T
P M R E P T E I R T S T K S B A E I L L
P O B P E N T T E W C D K B I K I T F U
O D K R R E E P S A J E O K J N T N E M
L U D A D L O O N D Q D L F T E O A S E
F L M E R A A C I T E V L E H I S T S X
Q A Q L I V Q D C M B L R I S T F S Q F
N T Z C V X H J G U L F Z H A I A Q X F
U I W U E S Z A T O A H N R H D L O C K
S O H K H B M Q Q C G R T S Y O Y E M I
Y N C S O U V L E C E K Y G E S C A P E

WORD LIST

ANALYSIS	LOGICAL
ANTIC	MEMORY
BREAK	MODULATION
BUSS	MULTIPLEXING
CLEAR	OPTION
COLD	PERIPHERAL
CONTROL	PROGRAM
DATA	RESET
DELETE	SECTOR
ESCAPE	SELECT
FLOPPY	SHIFT
HELVETICA	SPACE
HYPERDRIVE	START
INSERT	SYSTEM
INTERFACE	VALENTINE

GAMBITS

by Arlan Levitan

REAR GUARD - Adventure International

This tour-de-force arcade exercise of player-missile graphics combines extremely well done sound with nice horizontal scrolling, and multiple skill levels of play. The object of the game is to eliminate fleets of sixty Cyborgs that attempt to elude the player two at a time. If you allow more than ten Cyborgs per fleet to slip past you or run into the ground or a Cyborg after losing all of your shields, the construction crew you are defending is eliminated and the game ends. Cyborgs are usually destroyed by slipping above or below, and then behind them to fire. You can also stop the enemy by ramming, but each ram will cost you a shield. The size of the dart missile your ship fires varies with the skill level. So does the speed, debris created after being hit, and the offensive potential of the Cyborgs. Your shields are partially replenished after surviving a full fleet attack. One to four players may play at different skill levels but must pass the same joystick around, a minor irritation that novice programmers may wish to correct on their own. Congratulations to Neil Larimer for a balanced and enjoyable game that rates three out of four stars. A nice touch in the disk version is an arcade-like "hall of fame" for the ten best scores to date. A couple of minor complaints: After the game ends, the last skill factor played by a single player should be retained instead of resetting to level one. Also, what in the world is the reasoning behind having the user RUN "D:START" in the disk version when all that file contains is the statement RUN "D:GUARD.DSK". If there is any reason for such tomfoolery I'd like to hear it. My last complaint is that I never got too far in trying to score well in this game. How could a dedicated Trekkie ever shoot at the occasional Cyborg that looks too much like the Enterprise for comfort!?

P.S. THE FIRST FEW COPIES OF THE DISK VERSION AND POSSIBLY THE CASSETTE CONTAIN A MINOR ERROR THAT CAN LOCK UP THE GAME OCCASIONALLY. To fix this problem change the last statement in line 81 from "IF MS=A10 THEN 85" to "IF MS=A10 THEN 86" if it has not been corrected in the copy you buy. Thanks to Dan Horn from AI for helping me find the bug in our pre-release review copy and a big thanks to Scott Adams and crew for remaining one of the few software houses to recognize the software purchaser's need for archiving, and for supporting the Atari machine from the beginning and sticking with it.

QUICK BITS OUT OF THE BUCKET**WARLOCKS REVENGE - Synergistic Software**

A nicely done hi-res adventure translated from Applesoft for the Atari. Although the command vocabulary is rather limited, the ability to switch from one type of alter ego to another adds enough challenge to keep things interesting. Most screens are rather simple, but a few are truly stunning. Two disks full of adventure in over 100 screens will keep most busy for a while here.

MATCH RACER - Gebelli Software

Nasir Gebelli, the reigning king of Apple arcade style games turns his talents toward the Atari and gives a fairly impressive first performance. This one or two player race through various types of terrain is a bit predictable after you get used to the format of the seven or eight types of vertically scrolling screens. Watch for solid new releases from this direction. continued

ALI BABA AND THE FORTY THIEVES - Quality Software

The closest thing yet to Dungeons and Dragons for the Atari. Up to seventeen simultaneous characters join with Ali in his search for the sultan's daughter. A well designed input system makes keyboard input unnecessary after initial set-up. Good graphics, color, and sound make this a worthwhile addition to any D&D fan's library. A great game for groups.

MOUSKATTACK - On-Line Systems

John Harris, the gifted author of JAWBREAKER, decides to milk a lot of the code from his great PAC-MAN clone for another maze chase game. If you are a dedicated arcadian and haven't bought either program yet, look at the two and buy the one that appeals the most to you. Owning both is overkill in my biased opinion...spend the other thirty bucks and change on CAVERNS OF MARS from the APX instead.

PROTECTOR - Synapse Software

An angry Mike Potter recently kissed the ambitious John and Patty Bell of Crystalware goodbye and took his popular Defender derivative with him. The Synapse release is cleaner, prettier, and cheaper than the now unavailable Crystalware release. Mike has taken his share of brickbats in the past from us, but his coding does seem to be improving dramatically. This could be the avatar of great things yet to come from a rising software star. Try this before you buy to make sure its your cup of tea. The higher levels of play are rather ludicrous (impossible).

KAYDS - Computer Magic

Never have so many paid so much for so little. Nice packaging does not justify the high price commanded here. Lacking originality in execution and concept. Find a friend with a copy of John Palevich's SHOOT instead.

DODGE RACER - Synapse Software

A nicely done version of the popular arcade game HEAD-ON. Multiple players and options coupled with smooth action make this another winner for Synapse. The option which lets your opponent control the jam car is nice.

CRUSH CRUMBLE & CHOMP - Automated Simulations

Assume the alter ego of your favorite movie monster and wreak havoc upon the uncivilized worlds of Tokyo, New York, San Francisco, or Washington D.C. Plays rather slow but is well thought out and thoroughly enjoyable. Numerous options and commands make this complex simulation of the Monster Mash a thinking creatures delight.

REVERSAL - Hayden Software

Not as lightning fast as Artsci's Reversi but an abundance of options such as take back a move, displaying previous moves and a 2 player monitor mode make this the Othello look-alike of choice.

WIZARD & THE PRINCESS - On-Line Systems

A Hi-Res Adventure requiring wit, diligence, and careful observation. Worth the \$32.95 price of admission for seasoned adventurers.

POOL 1.5 - Innovative Design Systems Inc.

A beautifully done simulation of pocket billiards and its variations without the smoke-filled room and hustlers. The 48K it requires indicates a lot of brute force code here, but in this case, might makes right. Excellent ball action, variable english, ball acceleration and surface friction. One to four players - this is one of my favorites.

M.A.C.E.

Tutorial

Cassette I/O - Part 2

Using ENTER and LIST

or

How To Make Your Atari Write Programs

by Tom Giese

The ENTER command is used to input Basic programs and parts of programs from files generated from the LIST command. ENTER is also used to input Basic program statements generated by other programs. When using ENTER, tokenized Basic lines are created from the source statements in the input file.

The format for the ENTER command is:

ENTER filespec

Example:

ENTER "C:"

ENTER "D:LISTFILE.TMP"

When a line is ENTERed with the same line number as a currently existing line, the current line is replaced by the ENTERed line. If a Basic statement is ENTERed without a line number, the line is executed as in direct mode. When the end of the input file is reached, the input file is closed. When using ENTER within a program, the program ENds execution after the input file is closed. ENTER uses input/output control block #7.

The LIST command is used to output the source statements of the Basic program currently in memory. LIST creates source statements from the tokenized Basic program in memory.

The format for the LIST command is:

LIST filespec,lineno,lineno

Example:

LIST "C:",10,100

LIST "D:LISTFILE.TMP"



If no line numbers are specified, then the entire Basic program in memory will be LISTed. If only a single line number is specified, then only that line will be listed. If no filespec is given, the output will be directed to the display screen. If the range of lines contains no program lines, no output will be created. LIST uses input/output control block #7.

Here is a short Basic program to help in entering DATA statements.

```
0 DIM A$(1),B$(120),UP$(1):UP$=CHR$(28):POKE 82,2:GRAPHICS 0:? "Create Data Statements"
1 TRAP 1:? "Enter, Save,, Review, Change ";:INPUT A$:IF A$="S" THEN ? "File":;:INPUT B$:LIST B$,10,32767
3 IF A$="C" THEN ? "Change line ";:INP UT F:? CHR$(125):LIST F:? "CONT":POSIT ION 2,0:END
4 IF A$<>"E" THEN 1
5 ? "Start ";:INPUT F:? "Increment ";:INPUT I:T=F
6 ? T;" DATA";:INPUT B$:IF B$="" THEN ? UP$::GOTO 1
7 Z=INT((LEN(STR$(T))+LEN(B$)+45)/38):FOR X=1 TO Z:? UP$::NEXT X:? T;" DATA ";B$:? "CONT"
8 FOR X=-2 TO Z:? UP$::NEXT X:POKE 842,13:END
9 POKE 842,12:T=T+I:? UP$::GOTO 6
```



CRYPTOGRAM by Jerry Aamodt

This program demonstrates another way to use STRINGS in programs. In this case, we use them to solve cryptograms. To start, let's DIM our STRINGS and the matrix. I used 300 for A\$, B\$, and C\$ to provide ample space for long phrases.

```
10 DIM A$(300),B$(300),C$(300),D$(1),E$(1),A(26)
20 FOR I=1 TO 26:A(I)=-1:NEXT I:C=65:B$="":FOR I=1 TO 26
30 B=INT(26*RND(0)+1):IF A(B)<>-1 THEN 30
40 A(B)=C:C=C+1:NEXT I:RESTORE 1000+INT(6*RND(0)):READ A$:B$=A$
```

In line 20, we initialize our matrix, then begin creating the "code" to be used. This is done by selecting one of the 26 matrix locations and filling it with the ASCII value of "A", then "B", etc. We do this until all 26 matrix locations are filled with an ASCII value of the alphabet.

Now that the code is completed, we randomly select a phrase(A\$) from the DATA statements at the end of the program. We set B\$ equal to A\$ and begin encoding the phrase.

```
50 FOR I=1 TO LEN(A$):IF ASC(A$(I,I))>64 AND ASC(A$(I,I))<91
THEN B$(I,I)=CHR$(A(ASC(A$(I,I))-64))
60 NEXT I:?"":C$=B$:POSITION 2,6:?" C$:POSITION 2,15:?" B$
```

In line 50, the letters in B\$ are replaced, one by one, with the code contained in the matrix set up in line 30 and 40. We condition the program to only look at capital letters, ignoring all symbols or other characters. The program could be easily modified to look at lower case letters. Once encoded, we set the working STRING (C\$) equal to B\$, and print the two coded phrases on the screen.

```
100 POSITION 2,20:?"ENTER LETTER TO BE CHANGED";:INPUT D$
110 POSITION 2,20:?"ENTER LETTER TO BE TRIED ";:INPUT E$
120 FOR I=1 TO LEN(A$):IF B$(I,I)=D$ THEN C$(I,I)=E$
130 NEXT I:POSITION 2,6:?" C$:IF C$=A$ THEN 20
140 GOTO 100
```

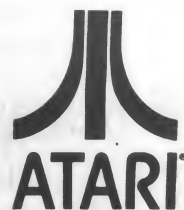
Lines 100 to 140 are the solving loop, prompting the player to select a letter, then substituting that letter in C\$. B\$ will remain unchanged at the bottom of the screen for reference. Line 130 checks C\$ to determine if it is the same as the randomly selected phrase we started with. If so, go to a win routine and randomly select a new phrase.

To complete the program, add DATA lines containing your choice of phrases. Change the random selector in line 40 to reflect the actual number of DATA lines. (1000+INT(number of DATA statements*RND(0)))

```
1000 DATA NOW IS THE TIME FOR ALL GOOD MEN TO DECODE
1001 DATA YOU'VE COME A LONG WAY BABY
1002 DATA SING A SONG OF SIXPENCE POCKET FULL OF RYE
1003 DATA I'VE NOT YET BEGUN TO FIGHT
1004 DATA ATTEND MACE MEETINGS AND LEARN MORE ABOUT YOUR COMPUTER
1005 DATA SUPPORT OUR ADVERTISERS; THEY MAKE THIS NEWSLETTER POSSIBLE
```


While They Last

New Arrivals for Atari



Software by Avalon Hill, Broderbund, BudgeCo, Crystalware,
Creative Computing, Epyx and more!

Disks: Maxwell

Magazines: Byte, Interface Age,
Creative Computing,
Compute

SERVICE

BYTE by BYTE
Microcomputers & Software

Classes in Programming — Sign Up Now

52070 Van Dyke 254-2698

Metro Plaza, between 23 & 24 Mile Rd.



Utica

WRITE ON!

By Richard Gizinski

Life on Baker Street isn't always easy with an editor like Marshall Dubin. Writing an article — that was easy. I had found out a few very interesting things about my Atari and I wanted to share them. Thus began the Baker Street Bytes. Now Marshall has given me a new request. Get other people to do the same. This is one of the problems that has plagued M.A.C.E. from the beginning.

Probably the most difficult thing in trying to share information is putting down the words you really want to say. You look back on what you've just written and say, "Whoops! I didn't really mean it that way." Or, "I should have mentioned that first. Now I have to start all over." When you think you've finally got it, oh oh, another goof. Boy! One of those great word processor programs would really be nice! Even a simple one would help. But you've just spent all your money getting your Atari and you wanted to play a few games too! What's a poor would-be writer to do.

and his wonderful word processor. Or mail it to M.A.C.E., P.O. Box 2785, Southfield, MI 48037. He'll return the media at the next monthly meeting.

Atari has a built in answer. You can use REM and DATA statements to hold text. You can write almost anything just by putting a line number and REM down first. You then have three lines to write in. Another line number and REM, another three lines. Clumsy, but it works. If you space your line numbers 10 apart, you can add or change lines just as you would in a program. Start the line numbers at 1000. Then you can overwrite the line numbers with numbers a thousand higher and change the REMs to PRINT statements. That way you can see what you've written in continuous order. After each paragraph, add a blank line. This will make it easier to see the paragraph breaks. Then CSAVE it (or save to disk) and give it to Marshall Dubin

MICHIGAN ATARI COMPUTER ENTHUSIASTS

President: Arlan Levitan
12709 Borgman
Huntington Woods, MI 48070

Vice Pres.: Jerry Aamodt
4148 Huhn
Rochester, MI 48063

Treasurer: Judy Braun
39281 Eliot
Mt. Clemens, MI 48043

Secretary: Sheldon Leemon
14400 Elm
Oak Park, MI 48237

Programs: Gary Luzier
22530 Gordon
St. Clair Shores, MI 48081

Librarians: Sam Findley (Disk)
8171 Denwood, Apt. #1
Sterling Heights, MI 48077

Rodney Graham (Tape)
12270 Deming
Sterling Heights, MI 48077

Newsletter: Marshall Dubin
2639 Hempstead
Auburn Heights, MI 48057

PROGRAM LIBRARY SUBMISSION FORM

I wish to submit the following program to the M.A.C.E. program library. I warrant that I have good title to this program, and that it does not infringe upon any copyright. Limited rights are transferred herewith for the use of this program within the membership of M.A.C.E. and its associated Atari Computer organizations.

Program Name _____ Submission Media: Tape _____ Disk _____ List _____
Description _____
Category: Business _____ Demonstration _____ Education _____ Game _____ Utility _____ Other _____
Minimum System: Memory _____ K BASIC _____ Assembler _____ PILOT _____ PASCAL _____
Disk _____ Tape _____ Interface _____ Modem _____ Printer _____ Light Pen _____
Joystick Controllers _____ Paddle Controllers _____ Keyboard Cont. _____
Name _____ Phone _____ Date _____

M.A.C.E. MEMBERSHIP APPLICATION

\$15.00 fee enclosed

Name _____ Phone _____
Street _____
City _____ State _____ Zip Code _____
Company (if applicable) _____
System Description _____ Disk/Tape _____
Suggestions _____
I can help with . . . _____

Send to: M.A.C.E.
P.O. Box 2785, Southfield, MI 48037

WRITE ON!

Next is the problem of what to write about. For many M.A.C.E. members, Atari is their first experience with a computer. Many beginners may share the same problems

What to write about? Almost anything! As a beginner, you can write about the trials and tribulations of getting past CLOAD and RUN. Others will take heart that they are not alone in entering the new world of home computing. Questions are always welcome for they give more insight into what type of articles would be helpful.

I've really appreciated articles like Sheldon Leemon's Down Memory Lane. Or Craig Chamberlain's Hyperwarp. Or Marshall Dubin's article on hooking your Atari to the outside world. There are lots of things left to write about from entry level to advanced. How about a disk enthusiast explaining the difference between fast and slow formatting. Some other topics you might like to explore are: The 'C' option of DOS, and its practical uses; A Glossary of Computer Terms; Using RESTORE and DATA statements; Disk Drives and how to get the most out of them; Good Techniques in writing a program. The list seems endless. Even short tips and aids like cutting a notch on the other side of a disk so that you can use the 'flip' side, or gripes like the program that you bought and it didn't...

This gets us down to ... how to go about writing it down. That's not as bad as you might think. If you keep in mind that the M.A.C.E. readership are friends that would like to know what you've found out. Just write it down like you would like to say it. Put one sentence to a REM or DATA statement and you can go back and correct or add to. If you make any boo boos, well, that's the job of the editor you've CSAVED the tape for. He can always take the blame.

Atari is a reg.™ of Atari, Inc.

UPGRADE YOUR ATARI®

REPORT CARD

STUDENT *Basic A+*

PARENT *Optimized Systems Software*

SUBJECTS

ADVANCED I/O	A+
STRUCTURED PROGRAMMING	A+
PLAYER/MISSILE GRAPHICS	A+

DEPARTMENT

EASE OF USE	A+
FLEXIBILITY	A+
COMPATIBILITY	A+

THE ONLY LOGICAL UPGRADE FOR ATARI BASIC!

GRADUATE TO BASIC A+ FROM THE AUTHORS OF ATARI BASIC!

BASIC A+

BASIC A+ will rate an A+ from any Atari user! Upward compatible with Atari Basic, it adds statements and features that enhance the Atari 800's real power, flexibility, and ease of use: Superior I/O features for business and other applications. Additional file manipulation commands. Significant help in program development and debug. Structured programming aids. And MORE! A partial list of the enhancements of BASIC A+ includes:

RPUT/RGET (record I/O) BPUT/BGET (binary I/O) ERASE
PRINT USING SET TAB INPUT "... " DIR PROTECT RENAME
TRACE WHILE...ENDWHILE IF...ELSE...ENDIF
SUPERB PLAYER/MISSILE GRAPHICS

BASIC A+ requires a disk and 32K bytes of RAM. Since no cartridge is used, BASIC A+ will take advantage of all the RAM (48K bytes) in a maximum Atari 800 system (recommended) \$80

OS/A+

Completely compatible with Atari's DOS (version 2), but with an advanced, command-driven console processor. Simple. Flexible. Powerful. With an easy-to-use BATCH capability. OS/A+ INCLUDES all the following utilities (and more):

EASMD (Editor/ASSEMBler/Debug) is our upgraded all-in-one assembly language development package for the 6502 microprocessor. The editor — with such features as FIND and REPLACE — can even edit BASIC A+ programs. The assembler can include multiple source files in a single assembly.

DUPDSK and FORMAT create master or slave disks. Make sector by sector copies of any OS/A+ disk.

COPY a simple, single file copy utility.

All of this power is included in our OS/A+ package \$80

For the utmost in capability and flexibility, our combination system — BASIC A+ and OS/A+ — is available \$150

SOFTWARE IN A CLASS BY ITSELF
AVAILABLE NOW THROUGH LOCAL DEALERS

Optimized Systems Software, Inc.
10379 Lansdale Ave., Cupertino, CA 95014
(408) 446-3099



MICHIGAN ATARI COMPUTER ENTHUSIASTS
P.O. BOX 2785
SOUTHFIELD, MICHIGAN 48037

BULK RATE
U.S. POSTAGE
PAID
PERMIT #431
SOUTHFIELD, MI

01/83

MEETING SCHEDULE, 1982

JANUARY 21	JULY 15
FEBRUARY 18	AUGUST 19
MARCH 18	SEPTEMBER 16
APRIL 15	OCTOBER 21
MAY 20	NOVEMBER 18
JUNE 17	DECEMBER 16

BERKLEY COMMUNITY CENTER
2400 ROBINA
BERKLEY, MICHIGAN

LOCATED OFF CATALPA (BETWEEN 11 AND 12 MILE)
AND BETWEEN GREENFIELD AND COOLIDGE

IMPORTANT DATED MATERIAL
PLEASE DO NOT DELAY